

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Título do trabalho**  
***um subtítulo***

Antonio Marcos Shiro Arnauts Hachisuca

MONOGRAFIA FINAL  
MAC 499 — TRABALHO DE  
FORMATURA SUPERVISIONADO

Supervisor: Prof. Dr. Marcel Kenji de Carli Silva

São Paulo  
2024

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0  
(Creative Commons Attribution 4.0 International License)*

*Esta seção é opcional e fica numa página separada;  
ela pode ser usada para uma dedicatória ou epígrafe.*



[illegible]



## Resumo

Antonio Marcos Shiro Arnauts Hachisuca. **Título do trabalho:** *um subtítulo*. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2024.

[illegible]

**Palavras-chave:** Palavra-chave1. Palavra-chave2. Palavra-chave3.





# Abstract

Antonio Marcos Shiro Arnauts Hachisuca. **Title of the document: *a subtitle***. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2024.

[illegible]

**Keywords:** Keyword1. Keyword2. Keyword3.



## Lista de abreviaturas

URL	Localizador Uniforme de Recursos ( <i>Uniform Resource Locator</i> )
IME	Instituto de Matemática e Estatística
USP	Universidade de São Paulo

## Lista de símbolos

## **Lista de figuras**

## **Lista de tabelas**

## **Lista de programas**

# Sumário

<b>Introdução</b>	<b>1</b>
<b>1 Blossom algorithm</b>	<b>3</b>
1.1 Matchings . . . . .	3
1.2 Maximum matching problem . . . . .	4
 <b>Índice remissivo</b>	 <b>7</b>



# Introdução

Alguma coisa.





# Capítulo 1

## Blossom algorithm

The main algorithm<sup>1</sup> to solve the **maximum matching** problem is the algorithm **Blossom** developed by Jack Edmonds in 1965. Hence, this chapter intends to: (1) introduce graph concepts relevant to solving the maximum matching problem; (2) Prove the Blossom algorithm; (3) Implement the Blossom algorithm and (4) present applications of the Blossom algorithm.

### 1.1 Matchings

**1.1.1 Graph:** A graph  $G$  is a triple  $(V, E, \varphi)$  such that

- (i)  $V$  is the **vertex set**;
- (ii)  $E$  is the **edge set**;
- (iii)  $\varphi : E \rightarrow V \times V$  is a relation between each edge and a pair of vertices, called the **incidence function** of  $G$ .

Usually, it is used  $V(G)$  or  $V_G$  to denote  $V$  and  $E(G)$  or  $E_G$  to denote  $E$ . Also, if  $e \in E(G)$  and  $\varphi(e) = (u, v)$ , then  $u$  and  $v$  are the **ends** of  $e$ ; When the context is clear,  $(u, v)$  may be abbreviated to  $uv$ .

**1.1.2 Subgraph:** A graph  $H$  is a **subgraph** of a graph  $G$  if  $V_H \subseteq V_G$ ,  $E_H \subseteq E_G$  and every edge in  $E_H$  has the same ends in  $H$  and  $G$ .

**1.1.3 Walk:** For a graph  $G := (V, E, \varphi)$ , a **walk** is a sequence

$$\langle v_0, e_1, v_1, \dots, e_l, v_l \rangle =: W$$

such that

- (i)  $l \in \mathbb{N}$  is the *length* of  $W$ ;
- (ii)  $v_0, v_1, \dots, v_l \in V$ ;

---

<sup>1</sup> If the graph is **guaranteed** to be bipartite, one may search *Kuhn's algorithm* and *Flow Networks*.

(iii)  $e_1, \dots, e_l \in E$ .

It is denoted that  $V(W) := \{v_0, \dots, v_l\}$  and  $E(W) := \{e_1, \dots, e_l\}$ . It is said that  $W$  is walk from  $v_0$  to  $v_l$  or a  $(v_0, v_l)$ -walk. The walk  $W$  is a:

- **path**, if all its vertices are distinct;
- **cycle**, if  $v_0 = v_l$  and it is an **odd cycle** if its length is odd, else it is an **even cycle**.

A vertex  $u \in V$  **reaches**  $v \in V$  if there is a  $(u, v)$ -walk in  $G$ .

**1.1.4 Components:** A **(connected) component** of  $G$  is a subgraph  $H$  such that every vertex of  $V_H$  reaches every vertex of  $V_H$ , but does not reach any vertex in  $V_G \setminus V_H$ . If  $G$  has **exactly** one component, then  $G$  is **connected**; Else,  $G$  is **disconnected**.

**1.1.5 Bipartite graphs:** A graph  $G$  is **bipartite** if there are two sets  $U, W \in V(G)$  such that

- (i)  $U \cap W = \emptyset$ ;
- (ii)  $U \cup W = V(G)$ ;
- (iii) every edge of  $G$  has one end at  $U$  and the other end at  $W$ .

In this case, it is said that  $G$  is  $(U, W)$ -bipartite.

**Theorem 1.1.6 (Characterization of bipartite graphs):** *A graph is bipartite if and only if it has no odd cycles.*

*Proof.*

□

**1.1.7 Matching:** For a graph  $G := (V, E, \varphi)$ , a set  $M \subseteq E$  is a **matching** of  $G$  if and only if no two edges in  $M$  share an end. A vertex  $v \in V$  is  $M$ -covered if some edge of  $M$  incides in  $v$ , and it is said that  $M$  covers  $v$ ; Otherwise,  $v$  is  $M$ -exposed. The matching  $M$  is:

- **maximal**, if there is no edge  $e \in E \setminus M$  such that  $M \cup \{e\}$  is a matching of  $G$ ;
- **maximum**, if for every matching  $M'$  of  $G$  one has  $|M| \geq |M'|$ ;
- **perfect**, if  $2|V_G| = |M|$ , i.e., every vertex of  $G$  is covered.

Denote  $\nu(G)$  as the size of a maximum matching in  $G$ .

## 1.2 Maximum matching problem

Now, the maximum matching problem can be described as:

### MAXMATCHING

*Given a graph  $G := (V, E, \varphi)$  find a maximum matching of  $G$ .*

**1.2.8 Alternating and augmenting paths:** Given a matching  $M$  of a graph  $G$ . A path  $P$  is  $M$ -**alternating** if its edges are alternating in and out of  $M$ . Formally,

$$e_i \in M \iff e_{i+1} \notin M \text{ for each } i \in [l-1]^2$$

And  $P$  is  $M$ -**augmenting** if both  $v_0$  and  $v_l$  are  $M$ -exposed.

**Theorem 1.2.9 (Berge's theorem):** Let  $G := (V, E, \varphi)$  be a graph. A matching  $M$  is maximum if and only if there are no  $M$ -augmenting path.

*Proof.* Let  $G := (V, E, \varphi)$  be a graph and  $M$  be a matching of  $G$ .

*Sufficiency:* It will be proven by contradiction. Suppose  $M$  is a maximum matching and  $P$  is an  $M$ -augmenting path of  $G$ . Note that, for  $i \in [l]$ , one has: (1)  $e_i \in M$ , if  $i$  is even; (2)  $e_i \notin M$ , if  $i$  is odd.

Let  $M' := M \Delta E(P)$ ,  $M'$  is a matching since  $v_0, v_l$  are  $M$ -exposed and every vertex in  $\{v_1, \dots, v_{l-1}\}$  is covered by an edge in  $M \cap E(P)$ . Hence,  $|M'| = |M| + 1$ , a contradiction.

*Necessity:* Suppose  $G$  has no  $M$ -augmenting paths. Let  $M'$  be a maximum matching of  $G$  and  $G'$  be the graph induced by  $M \Delta M'$ . Note that  $G'$  has at least one component and every component of  $G'$  is either a path or a cycle. Let  $H$  be a component of  $G'$ ,

1. if  $|V_H|$  is **even**, then  $|M \cap V_H| = |M' \cap V_H|$ ;
2. if  $|V_H|$  is **odd**, then  $H$  must be a path. Note that  $M'$  being maximum implies  $|M \cap V_H| > |M' \cap V_H|$  and, consequently,  $M' \cap V_H$  is a  $M$ -augmenting path of  $G$ . Therefore, this case is impossible.

Thus,  $|M| = |M'|$ , i.e.,  $M$  is a maximum matching. □

**1.2.10 Vertex cover:** For a graph  $G := (V, E, \varphi)$ , a subset  $K \subseteq V(G)$  is a **vertex cover** of  $G$  if every edge of  $E(G)$  has an end in  $K$ . A vertex cover is said to be **minimal** if one has  $|K| \leq |K'|$  for every vertex cover  $K'$  of  $G$ . Denote  $\tau(G)$  as the size of a minimum vertex cover of  $G$ .

**Corollary 1.2.11 (Maximum matching upperbound):** Let  $G$  be a graph,  $M$  be a matching of  $G$  and  $K$  be a vertex cover of  $G$ . Then,  $|M| \leq |K|$ .

*Proof.* By definition, every edge of  $M$  has at least one end in  $K$  and no edge in  $M$  share an end. Hence,  $|K| \geq |M|$ . □

**Theorem 1.2.12 (König's matching theorem):** Let  $G$  be a bipartite graph, then the maximum size of a matching of  $G$  is equal to the minimum size of a vertex cover of  $G$ .

*Proof.* □

Note that König's theorem **does not** hold for all graphs; It suffices to consider a single odd cycle.

---

<sup>2</sup> For  $n \in \mathbb{N}$ , we denote the set  $\{1, \dots, n\}$  as  $[n]$ .



# Índice remissivo

## C

Captions, *veja* Legendas

Código-fonte, *veja* Floats

## E

Equações, *veja* Modo matemático

## F

Figuras, *veja* Floats

Floats

Algoritmo, *veja* Floats, ordem

Fórmulas, *veja* Modo matemático

## I

Inglês, *veja* Língua estrangeira

## P

Palavras estrangeiras, *veja* Língua es-

trangeira

## R

Rodapé, notas, *veja* Notas de rodapé

## S

Subcaptions, *veja* Subfiguras

Sublegendas, *veja* Subfiguras

## T

Tabelas, *veja* Floats

## V

Versão corrigida, *veja* Tese/Dissertação,  
versões

Versão original, *veja* Tese/Dissertação,  
versões